Remarks

Reconsideration of this Application is respectfully requested.

Upon entry of the foregoing amendment, claims 1-20 are pending in the application, with 1, 6, 9, 14 and 20 being the independent claims. Claims 1, 6, 9, 14 and 20 are sought to be amended. These changes are believed to introduce no new matter, and their entry is respectfully requested.

Based on the above amendment and the following remarks, Applicants respectfully request that the Examiner reconsider all outstanding rejections and that they be withdrawn.

Telephone Interview with the Examiner of March 7, 2003

Applicants' representative thanks the Examiner for the courtesies extended during the telephone conference of March 7, 2003. Pursuant to the Examiner's request, Applicants submit this amendment and the remarks in writing.

Rejections under 35 U.S.C. § 102

Claims 1 and 14 stand rejected under 35 U.S.C. § 102(b) as being allegedly anticipated by Buzbee, U.S. Patent No. 5,815,720. As discussed during the telephone interview, there are a number of differences between the invention recited in claim 1, and Buzbee. First, claim 1 recites that the "first intermediate representation includes instructions instrumented into the *source* code." Buzbee, on the other hand, teaches instrumenting instructions into the *object* code. See Buzbee, col. 6, lns. 4 - 17.

Additionally, claim 1 recites the use of *numerical* annotations (see claim 1, step 2, "annotating said first intermediate representation with previously gathered numerical feedback data"). The annotations of Buzbee, on the other hand, are not numeric annotations, but three-part code-like objects that are later translated into assembly-level instructions by the dynamic translator. See Buzbee, col. 7, ln. 52 -col. 8, ln. 12. Thus, Applicants respectfully submit that although the same word ("annotation") is used, the meaning is clearly different, in view of the present specification (see, for example, page 14, line 16 - page 16, line 7 of the present specification).

Furthermore, a careful review of Buzbee shows that Buzbee performs its optimization *during multiple compile passes*. See Buzbee, col. 5, lns. 50 - 52. This is the opposite of what is claimed in claim 1, which recites "repeating steps 3 and 4 at least once *during the same compilation pass*." Applicants further submit that this limitation was already present in claim 1 (see the prior language "during compile-time"), but, Applicants have amended this language to more clearly point out that the *multiple* performance of the updating step and the optimization step occurs during the same compilation pass.

Applicants submit that the above-noted amendments are being made to more clearly define the invention and are not intended to narrow the scope of claim 1 in any way.

Claim 14 has been similarly amended to clarify that the updating and the optimization steps are performed multiple times during the same compile pass. As discussed above with reference to claim 1, at least this aspect of claim 14 is not disclosed in Buzbee. Furthermore, as with claim 1, claim 14 recites instrumenting instructions into the *source* code, rather than into the *object* code (which Buzbee teaches).

Because not every element of claims 1 and 14 is taught by Buzbee, Applicants respectfully submit that Buzbee does not anticipate claims 1 and 14. Accordingly, Applicants respectfully request that the rejection of claims 1 and 14 under 35 U.S.C. § 102(b) be reconsidered and withdrawn.

Rejections under 35 U.S.C. § 103

Claims 2, 3, 4 and 5 stand rejected under 35 U.S.C. § 103(a) as being allegedly unpatentable over Buzbee in view of a number of references. Since claim 1 is allowable for the reasons discussed above, its dependent claims are allowable for at least the same reasons. Accordingly, Applicants respectfully request that the rejections under 35 U.S.C. § 103(a) of claims 2, 3, 4 and 5 be reconsidered and withdrawn.

Claims 15-19 stand rejected under 35 U.S.C. § 103(a) as being allegedly unpatentable over Buzbee in view of a number of references. Since claim 14 is allowable for the reasons discussed above, its dependent claims are allowable for at least the same reasons. Accordingly, Applicants respectfully request that the rejections under 35 U.S.C. § 103(a) of claims 15-19 be reconsidered and withdrawn.

Claim 6 stands rejected under 35 U.S.C. § 103(a) as being allegedly unpatentable over Buzbee. Claim 6 has been amended in a manner similar to claim 1. Applicants respectfully submit that claim 6 is allowable over Buzbee at least for the reasons applicable to claim 1, as discussed above, as well as due to the features recited therein.

Accordingly, Applicants respectfully request that the rejection of claim 6 under 35 U.S.C. § 103(a) be reconsidered and withdrawn.

Claim 7 and 8, which depend from claim 6, are allowable at least for the reasons applicable to claim 6, as well as due to the features recited therein.

Independent claims 9 and 20 have been similarly amended to more clearly point out the aspect of performing the updating and optimization steps multiple times during the same compilation pass. These claims are allowable at least for the reasons applicable to claim 1, as well as due to the features recited therein. Applicants further note that these claims expressly recite the global and local *frequency* data as annotations (i.e., they inherently recite that the annotations are numerical, rather than object code instructions of Buzbee).

Accordingly, Applicants respectfully request that the rejection of claims 9 and 20 under 35 U.S.C. § 103(a) be reconsidered and withdrawn.

Conclusion

All of the stated grounds of objection and rejection have been properly traversed, accommodated, or rendered moot. Applicants therefore respectfully request that the Examiner reconsider all presently outstanding objections and rejections and that they be withdrawn. Applicants believe that a full and complete reply has been made to the outstanding Office Action and, as such, the present application is in condition for allowance. If the Examiner believes, for any reason, that personal communication will expedite prosecution of this application, the Examiner is invited to telephone the undersigned at the number provided.

Prompt and favorable consideration of this Amendment and Reply is respectfully requested.

Respectfully submitted,

STERNE, KESSLER, GOLDSTEIN & FOX P.L.L.C.

George S. Bardmesser
Attorney for Applicants
Registration No. 44,020

Date: 3/21/03

1100 New York Avenue, N.W.
Suite 600
Washington, D.C. 20005-3934
(202) 371-2600

Version with markings to show changes made

In the claims:

Please amend claims 1, 6, 9, 14 and 20 as follows:

1. (Twice Amended) A method for precise feedback data generation and updating during compile-time optimizations, within an optimizing compiler, comprising:

(1) accessing a first intermediate representation of source code of a computer program, wherein said first intermediate representation includes instructions instrumented into the source code of said computer program;

(2) annotating said first intermediate representation with previously-gathered numerical feedback data from a plurality of sample executions of said computer program;

(3) updating said numerical feedback data [during compile-time] according to a pre-defined propagation scheme;

(4) performing an optimization of said first intermediate representation annotated with said numerical feedback data updated in step (3), thereby producing a transformed intermediate representation; and

(5) repeating steps (3) and (4) at least once during the same compilation pass[;

whereby the compiler produces more efficient executable program code from said first intermediate representation, thus speeding up execution of said computer program].

6. (Twice Amended) A computer program product comprising a computer usable medium having computer readable program code means embodied in said medium for causing an application program to execute on a computer that performs precise feedback

data generation and updating during compile-time optimizations, within an optimizing compiler, said computer readable program code means comprising:

first computer readable program code means for causing the computer to access a first intermediate representation of source code of a computer program, wherein said first intermediate representation includes instructions instrumented into the source code of said computer program;

second computer readable program code means for causing the computer to annotate said first intermediate representation with previously-gathered numerical feedback data from a plurality of sample executions of said computer program;

third computer readable program code means for causing the computer to update said feedback data [during compile-time] according to a pre-defined propagation scheme;

fourth computer readable program code means for causing the computer to perform an optimization of said first intermediate representation annotated with said numerical feedback data updated by said third computer readable program code means, thereby producing a transformed intermediate representation; and

fifth computer readable program code means for causing the computer to re-execute said third and fourth computer readable program code means at least once during the same compilation pass[;

whereby the compiler produces more efficient executable program code from said first intermediate representation, thus speeding up execution of said computer program].

9. (Amended) A method for compile-time optimization comprising:

(1) accessing a first intermediate representation of [souce] source code of a computer program, wherein the first intermediate representation includes instructions instrumented into the source code;

(2) annotating the first intermediate representation with previously-gathered global and local frequency data from a plurality of sample executions of the computer program;

(3) updating the global and local frequency data according to a pre-defined propagation scheme;

(4) performing an optimization of the first intermediate representation annotated with the global and local frequency data updated in step (3) to produce a transformed intermediate representation; and

(5) repeating steps (3) and (4) at least once during the same compilation pass.

14. (Amended) A method for compile-time optimization comprising:

(1) accessing a first intermediate representation of souce code of a computer program, wherein the first intermediate representation includes instructions instrumented into the source code;

(2) annotating the first intermediate representation with previously-gathered feedback data from a plurality of sample executions of the computer program;

(3) updating the feedback data according to a pre-defined propagation scheme at multiple points during a compilation process;

(4) performing an optimization of the first intermediate representation annotated with the feedback data updated in step (3) to produce a transformed intermediate representation; and

(5) repeating steps (3) and (4) at least once during the same compilation pass.

20. (Amended) A method for compile-time optimization comprising the steps of:

(1) accessing a first intermediate representation of souce code of a computer program, wherein the first intermediate representation includes instructions instrumented into the source code;

(2) annotating the first intermediate representation with previously-gathered estimated frequency data from a plurality of sample executions of the computer program;

(3) updating the estimated frequency data according to a pre-defined propagation scheme;

(4) performing an optimization of the first intermediate representation annotated with the estimated frequency data updated in step (3) to produce a transformed intermediate representation; and

(5) repeating steps (3) and (4) at least once during the same compilation pass.